Ollscoil na hÉireann, Gaillimh

National University of Ireland, Galway

Semester I Examinations 2008 / 2009

Exam Code(s)	4IF, 4BP, 3BA, 1SD
Exam(s)	4th Year B.Sc. (Information Technology) 4th Year B.E. (Electronic & Computer Engineering) 3rd Year B.A. (Information Technology) Higher Diploma in Applied Science (Software Design & Development)
Module Code(s)	CT404, CT336
Module(s)	GRAPHICS AND IMAGE PROCESSING
Paper No.	
Repeat Paper	
External Examiner(s)	Prof. J. Keane Prof. S. McClean
Internal Examiner(s)	Prof. G. Lyons Dr. S. Redfern
Instructions:	Answer **any three** questions. All questions carry equal marks.
Duration	**2 hours**
No. of Pages	7
Department(s)	Information Technology
Course Co-ordinator(s)	
Requirements:	
MCQ	
Handout	
Statistical Tables	
Graph Paper	
Log Graph Paper	
Other Material	

Q.1.

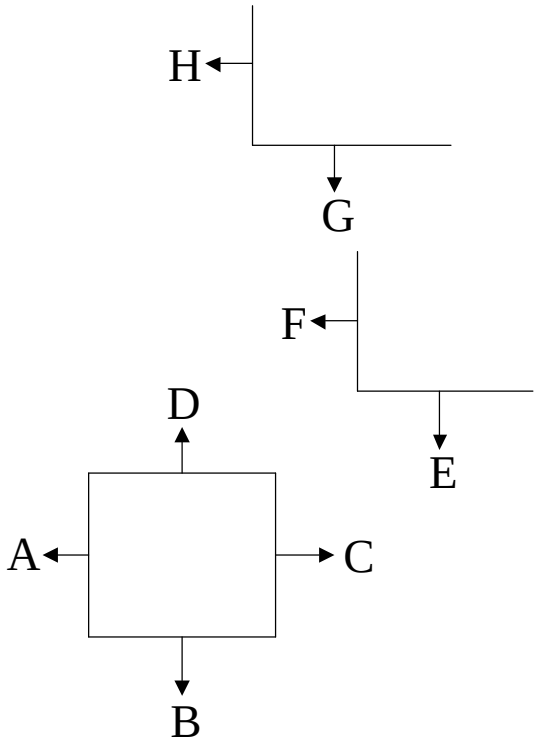
(a) What is a *surface normal*? (*2 marks*).

Discuss, with the use of a diagram in each case, the following 3D graphics techniques, and identify the importance of *surface normals* to each (*4 x 2 marks*):

i. Flat (Lambert) Shading
ii. Normal Interpolating (Phong) Shading
iii. Bump Mapping
iv. Back-Face Culling

(b) The *Binary Space Partitioning* (BSP) algorithm is widely used in modern graphics programming.

i. In what situation is the BSP approach most useful? In what situation is it not useful at all? (*2 marks*)
ii. Consider the diagram below, which depicts a simple 2D scene involving 8 polygons. The polygons are labeled A through H and the arrows indicate their *surface normals.* Construct a BSP tree for this scene, and briefly explain your steps in constructing it. (8 *marks*)



Q.2.

(a) *Antialiasing* is an approach in 2D raster graphics, which uses colour (depth) as a means to simulate an increase in resolution. Discuss the *antialisasing* technique, with specific reference to the concept of sub-pixel accuracy (*5 marks*).

(b) Why are *back buffers* used in real-time graphics applications? Explain their operation. (*4 marks*).

(c) *Extrusion*

(i) Describe the use of extrusion in VRML, referring to each of the seven fields used by the Extrusion node. (*5 marks*)

(ii) Write VRML code to produce an extruded shape, similar to the chess piece shown below. You should consider its geometry only, and can ignore the use of materials. Note that the most useful VRML nodes, as well as a cross section for this object, are summarised on the final page of this exam paper. (*6 marks*)

```
Extrusion
{
  crossSection [ ]
  spine        [ ]
  scale        [ ]
  orientation  [ ]
  beginCap
  endCap
  creaseAngle
}
```

Q.3.

(a) In order to set up the projection and camera in a 3D *OpenGL* application, the following two functions are often used:

```
gluPerspective(fov, aspect, near, far);
gluLookAt(eyex, eyey, eyez, atx, aty, atz,  upx, upy, upz);
```

Explain the use of these two functions, in particular the precise meaning of their arguments (*6 marks*).

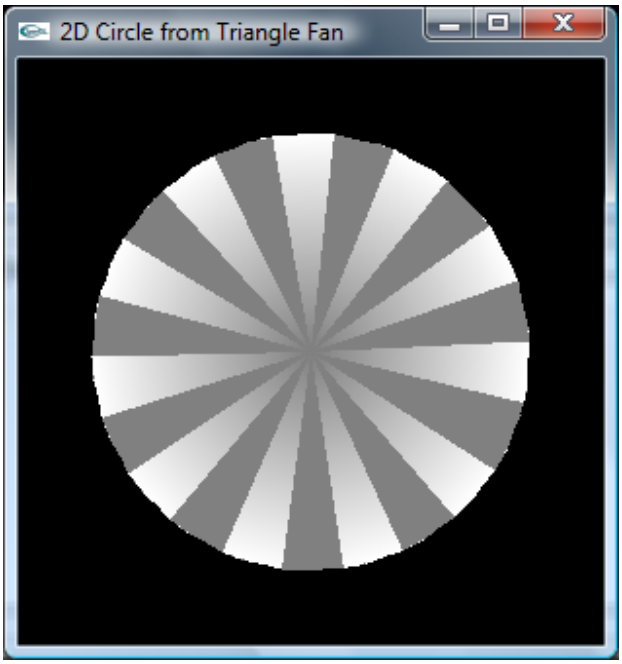
(b) The *OpenGL* program provided below draws a wire cube at the origin. Modify the program so that it uses an *idle callback* function to perform a simple repeating animated rotation of the cube (*10 marks*).

(c) Further modify the program so that it uses the `gluPerspective` and `gluLookAt` functions to define the projection and camera parameters. Ensure that the values you choose for the arguments to these functions are appropriate so that the cube is visible and centred in the window (*4 marks*).

```
#include <GL/glut.h>
void display();

int main(int argc, char** argv)
{
   glutInit(&argc,argv);
   glutCreateWindow("OpenGL Cube");
   glutDisplayFunc(display);
   glutMainLoop();
}

void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glFrustum(-1.0, 1.0, -1.0, 1.0, 1.0, 5.0);

   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   glutWireCube(1.0);
   glFlush();
}
```

Q.4.

(a) Write the `display()` function for an *OpenGL* program which renders the following 2D circle using the `GL_TRIANGLE_FAN` primitive. (Note the use of colour to differentiate the triangles in the fan) (*8 marks*):



(b) Explain the *keyframe* approach to animation in computer graphics, and explain its use in *VRML*, referring to the *TimeSensor*, *Transform*, *OrientationInterpolator* and *PositionInterpolator* nodes in your answer. (*6 marks*)

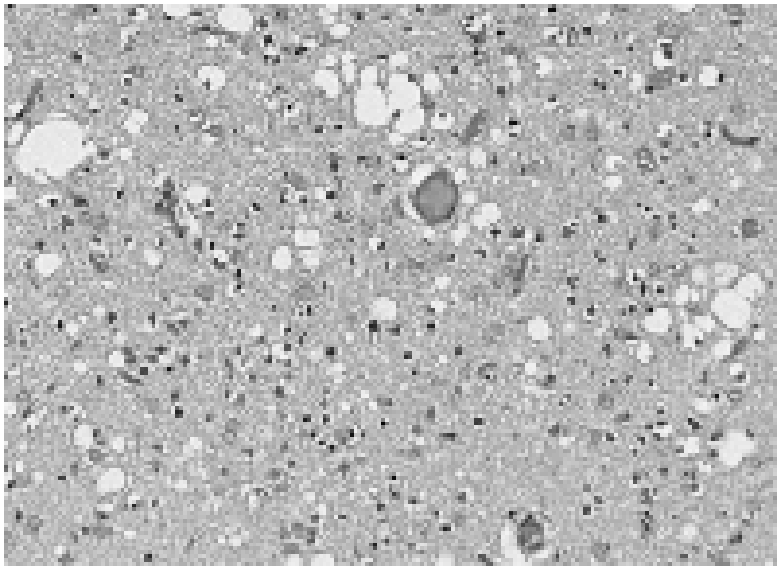
(c) A more powerful approach to producing animations in *VRML* is to use *JavaScript* nodes to dynamically calculate key positions or orientations. Write *VRML* code for a *JavaScript* node which produces a smooth elliptical motion, and which is suitable for use with a *TimeSensor* and a *Transform* node (*6 marks*).

Q.5.

(a) In digital image processing, what is meant by *histogram manipulation*? What is it used for? In your answer, refer to circumstances where two different types of histogram manipulation would be beneficial (*6 marks*).

(b) Describe the *mathematical morphology* approach to image processing. Outline some typical circumstances in which this approach is useful (*6 marks*).

(c) The image below is taken from tissue sample of a human brain affected by neurological damage. Of interest are white areas that are at least 5 pixels in diameter. **Outline** and **discuss** an algorithm for automatic isolation of areas matching this specification (*8 marks*).



Some useful VRML node information:

```
Shape
{
       geometry
       appearance
}

Transform
{
  children    [ ]
  translation 0.0  0.0  0.0
  rotation    0.0  0.0  1.0  0.0
  scale       1.0  1.0  1.0
  center      0.0  0.0  0.0
}

TimeSensor
{
  enabled            TRUE
  startTime          0.0
  stopTime           0.0
  cycleInterval      1.0
  loop               FALSE
  isActive           # eventOut
  time               # eventOut
  cycleTime          # eventOut
  fraction_changed   # eventOut
}

PositionInterpolator
{
  key   [  ]
  keyValue [  ]
  set_fraction   # eventIn
  value_changed  # eventOut
}

OrientationInterpolator
{
  key   [  ]
  keyValue [  ]
  set_fraction       # eventIn
  value_changed      # eventOut
}

Extrusion
{
  crossSection [ ]
  spine       [ ]
  scale       [ ]
  orientation [ ]
  beginCap
  endCap
  creaseAngle
}
```

```
Box
{
       size 2.0 2.0 2.0
}


Sphere
{
       radius              1.0
}

Cylinder
{
       radius              1.0
       height              2.0
       side                TRUE
       top                 TRUE
       bottom              TRUE
}

Appearance
{
       material
}

Material
{
       diffuseColor
       specularColor
       ambientIntensity
       emissiveColor
       transparency
       shininess
       texture
}

ImageTexture
{
       url
}
```

Co-ordinates for a circle-shaped cross section, suitable for extrusion:

```
1.00 0.00,    0.92 -0.38,
0.71 -0.71,   0.38 -0.92,
0.00 -1.00,   -0.38 -0.92,
-0.71 -0.71,  -0.92 -0.38,
-1.00 0.00,   -0.92 0.38,
-0.71 0.71,   -0.38 0.92,
0.00 1.00,    0.38 0.92,
0.71 0.71,    0.92 0.38,
1.00 0.00
```